REFERENCE

NBS
PUBLICATIONS

NAT'L INST. OF STAND & TECH
A11106 031619

NBSIR 85-3292

Release Notes for STAT2 Version 2.00A: An Addendum to NBS Special Publication 400-75

C. H. Ellenwood and R. L. Mattis

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Electronics and Electrical Engineering
Semiconductor Electronics Division
Gaithersburg, MD 20899

January 1986

QC
100
.U56
85-3292
1986
C.3

ARTMENT OF COMMERCE
BUREAU OF STANDARDS

NBSIR 85-3292

RELEASE NOTES FOR STAT2 VERSION 2.00A: AN ADDENDUM TO NBS SPECIAL PUBLICATION 400-75

C. H. Ellenwood and R. L. Mattis

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Electronics and Electrical Engineering
Semiconductor Electronics Division
Gaithersburg, MD 20899

January 1986

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

Release Notes for STAT2 Version 2.00A:

An Addendum to NBS Special Publication 400-75

Page

Abstract 1

Disclaimer 1

1. Introduction 1

2. Program Capabilities 2

3. Logical Unit Assignments 3

4. Program Installation 3

Acknowledgments 4

References 4

Appendix I. Source Listing of PCFG Program 5

Appendix II. Sample Run of PCFG Program 12

Appendix III. STAT2 Version 2.00A Commands 13

Release Notes for STAT2 Version 2.00A: An Addendum to NBS Special Publication 400-75

C. H. Ellenwood and R. L. Mattis
Semiconductor Electronics Division
National Bureau of Standards
Gaithersburg, MD 20899

Abstract

STAT2 is a FORTRAN program which is used to analyze and display data from microelectronic test structures fabricated on semiconductor wafers. The program reads data as a two-dimensional array, extracts sample statistical values, identifies outliers, calculates replacement values for outliers, and makes histograms and circular gray-tone data maps. Version 2.00A is an adaptation of STAT2 to run under Version 3.2 of the RSX-11M operating system. This operating system is used on the automatic tester which acquires the test structure data. Data can therefore be taken and analyzed on the same system.

Key Words: ATE; computer program; outlier; process validation wafer; statistical analysis; two-dimensional map; wafer map.

Disclaimer

Certain commercial equipment, instruments, or materials are identified in this report in order to adequately specify the experimental procedure. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the material or equipment identified is necessarily the best available for the purpose.

1. Introduction

This document describes the changes that were made to the STAT2 computer program documented in NBS Special Publication 400-75, Semiconductor Measurement Technology: A FORTRAN Program for Analysis of Data from Microelectronic Test Structures [1] and NBSIR 83-2779, Release Notes for STAT2 Version 1.31: An Addendum to NBS Special Publication 400-75 [2]. The changes were made in order to make STAT2 run under Version 3.2 of the RSX-11/M operating system. The referenced publications are to be used as the program manuals, for this document contains no review of STAT2 features or operation. Data base structure, functional fits, and the help facility are not included in this version of STAT2. This version was designed to contain the essential commands for statistical analysis of data taken on an Accutest 3000 automatic tester. The gray-tone map produced by the MP4 command is written specifically for a Printronix P300 or P600 line printer/plotter.

2. Program Capabilities

The STAT2 ASG and REA commands are used for reading input. Format 4 of the REA command is the only acceptable format which can be used with this version. Any other formats will give the error message "*** FORMAT TYPE MUST BE 4." The .SAD file asked for by the REA command is a file which gives the STEND array and the row-column coordinate locations of all untested sites within the region defined by the STEND array. This .SAD file is made using a stand-alone program called PCFG. A listing of PCFG is included in Appendix I. PCFG is a Pascal program which prompts the user for the beginning and ending columns of chips on each row of the wafer. It about the locations of the untested sites on the wafer and uses this information to generate the needed .SAD file. This program also generates the .PDF file and the .PID file which are needed to run a test plan on the automatic tester. A sample run of the PCFG program is shown in Appendix II.

PRS, DIS, LAP, LIP, and LNS, which give statistical information about the data, are available in version 2.00A. Also available are XOL, XPP, XNS, XGT, XLT, XIP, LXP, IIP, and RES, which deal with test site exclusion, as well as AXP and AIP which deal with data value replacement.

To meet the need to analyze small sample sizes, a new command, XSS (for exclude small sample), has been added to STAT2 which can be used to identify and exclude outliers when the number of included sites is in the range 3 to 25, inclusive. The format of the command is "XSS P1" where P1 represents the probability that one or more good sites might be excluded along with the outliers. P1 may be given values of 0.6, 0.4, 0.2, 0.1, 0.04, 0.02, or 0.01; however, the value of 0.2 was found to be reasonable for most data sets. Other values produce an error message.

The algorithm for the XSS command employs the Dixon test [3]. For this test, data values are first sorted and then tests are made on the extreme values. If one of the extreme values is excluded, a second examination is made based on the new extreme values. This process continues until no more values are excluded. Conceivably, with a sample size of only 3 or 4, XSS may exclude one or two sites, leaving fewer than the minimum three sites required. XSS will automatically calculate and print statistics upon completion. A typical command sequence might be as follows:

```
ASG FILE.DAT     ! Assign input data file.
REA 4 15         ! Read in the desired data set.
FILE.SAD         ! Specify .SAD file
XGT 1E10         ! Exclude values which are
XLT 0            !   known to be bad data or
XIP 1 6          !   to have physical defects.
XSS 0.2          ! Invoke XSS.
END              ! End of this example.
```

Additional information concerning the XSS command is contained in the introductory

comments to subroutine XSS in the source code.

The PLT, HIS, and MP4 commands are available to produce various types of data displays. The PLT command differs from previous versions in that the display is written to the STAT2.LOG file as well as to the terminal. The MP4 command differs from previous versions in that parameter P4, previously used to optionally specify the format of the map key, no longer performs that function. P4 must still be present; but regardless of its value, the numbers in the key print in F10.5 or E12.5 format depending on their magnitude.

Miscellaneous commands such as END, PAU, ERM, REM, and MAC still function as under previous versions.

The features which are not included in Version 2.00A are commands for functional fits, data base commands and the help facility. These commands give the error message "*** COMMAND NOT AVAILABLE" if entered. (These commands are still subjected to syntax checking even though they are not implemented.)

A list of commands which are available in Version 2.00A along with a brief description is given in Appendix III.

3. Logical Unit Assignments

The logical unit assignments used by STAT2 Version 2.00A along with a description are given below.

1 - Input data file
2 - Macro command file
3 - .SAD file
4 - STAT2.LOG file
5 - Command input
6 - Command echoes and error messages
7 - Scratch file used by the MP4 command

4. Program Installation

For STAT2 to run under the RSX-11/M operating system, it was necessary to divide the program into overlays. The code sections of STAT2 are: (1) ST2MAIN.FTN, (2) ST2OV1.FTN, (3) ST2OV2.FTN, (4) ST2OV3.FTN, and (5) ST2OV4.FTN. The code sections are compiled separately, and the task is built to form one program unit. The program requires approximately 30 KB of memory.

The main segment, ST2MAIN, contains the command interpreter, program initialization, and several subroutines that are needed by the program code in more than one overlay. The subroutines relating to the ASG, REA, MAC, and HIS commands are contained in ST2OV1. ST2OV2 includes subroutines relating to the test site exclusion and statistical calculations. ST2OV3 contains subroutines relating to the calculation and specification of replacement values for outliers. The subroutines relating to the PLT and MP4 commands are contained in ST2OV4.

ST2ACCU1.TKB is a task-building command file called by typing:

```
TKB @ST2ACCU1.TKB
```

The .TKB file consists of the following commands:

```
STAT2,STAT2/-SP=ST2ACCU1/MP
ACTFIL = 7
UNITS = 7
ASG = TI0:6,SY0:7
//
```

The first line of the .TKB file contains the output filenames and the name of the .ODL file. ST2ACCU1.ODL contains the names of the files needed by the task builder and defines the overlay structure. The form of ST2ACCU1.ODL is as follows, where LIBRARY.LIB represents a FORTRAN object library:

```
01:     .FCTR     ST2OV1-LIBRARY.LIB/LB
02:     .FCTR     ST2OV2-LIBRARY.LIB/LB
03:     .FCTR     ST2OV3-LIBRARY.LIB/LB
04:     .FCTR     ST2OV4-LIBRARY.LIB/LB
RT:     .FCTR     ST2MAIN-LIBRARY.LIB/LB
.ROOT RT-*(01,02,03,04)
.END
```

The ACTFIL option in the .TKB file declares the number of files that are simultaneously open in a task and the UNITS option declares the number of logical units that are used by the task. The ASG option makes the appropriate logical unit assignments.

Acknowledgments

The authors are indebted to Ruth Zucker for writing the PCFG program and to Loren W. Linholm for careful reading and helpful suggestions regarding the manuscript. The manuscript was prepared with proficiency by E. Jane Walters.

References

1. Mattis, R. L., Semiconductor Measurement Technology: A FORTRAN program for Analysis of Data from Microelectronic Test Structures, NBS Special Publication 400-75 (July 1983).

2. Mattis, R. L. and Zucker, R., Release Notes for STAT2 Version 1.31: An Addendum to NBS Special Publication 400-75, NBSIR 83-2779 (November 1983).

3. Natrella, M. G., Experimental Statistics, NBS Handbook 91, pp.17-3 and T-27 (Reprinted October 1966).

APPENDIX I

Source Listing of PCFG Program

```
PROGRAM PCFG (INPUT,OUTPUT,PIDFILE,EXTFILE,PDFFILE);
(*   CREATE A PROBER IDENTIFICATION AND DESCRIPTION FILE, .PID, .SAD AND .PDF*)

(**** REVISED FEBRUARY 1983 ****)


LABEL    100,250,300,350,400,500,600,700,800,900;

CONST    LSTEN=32;
         NDROPS=20;
         ONE=1;
         ZERO=0;


VAR      DROWS,DCOLS:INTEGER;
         FLAG,I,IA,I1,I2,J,JA,K,KL,LDROPS,LASTCOL:INTEGER;
         LOCATION,N,NCHAR,NCOLS,NROWS,NTEMP,NUMBER:INTEGER;
         L,TALLY,X:INTEGER;
         RTEMP:REAL;
         DROPS:ARRAY [1..20] OF ARRAY [1..2] OF INTEGER;
         DIESIZE:ARRAY [1..2] OF INTEGER;
         RDIE:ARRAY[1..2] OF REAL;
         RREF:ARRAY[1..2] OF REAL;
         REFERENCE:ARRAY [1..2] OF INTEGER;
         COMPDROPS:ARRAY [1..20] OF ARRAY [1..2] OF INTEGER;
         STEND:ARRAY [1..32] OF ARRAY [1..2] OF INTEGER;
         XLOC:ARRAY [1..1024] OF INTEGER;
         YLOC:ARRAY [1..1024] OF INTEGER;
         RXLOC:ARRAY [1..1024] OF REAL;
         RYLOC:ARRAY [1..1024] OF REAL;
         NAMEPID:PACKED ARRAY[1..20] OF CHAR;
         NAMEEXT:PACKED ARRAY[1..20] OF CHAR;
         NAMEPDF:PACKED ARRAY[1..20] OF CHAR;
         FILENAME:PACKED ARRAY[1..9] OF CHAR;
         PIDEXT,SADEXT,PDFEXT,DISKNAME:PACKED ARRAY[1..4] OF CHAR;
         BLANK:CHAR;
         PIDFILE,EXTFILE,PDFFILE : TEXT;

PROCEDURE PRINTINFO;
   BEGIN
      WRITELN ('    NROWS=', NROWS:3,  '    NCOLS=', NCOLS:3);
      WRITELN ('DIE SIZE=',DIESIZE[1]:5,  DIESIZE[2]:5);
      WRITELN ('REFERENCE=', REFERENCE[1]:5, REFERENCE[2]:5);
      WRITELN;
      WRITELN ('    I',  '     STEND[I,1]', '     STEND[I,2]');
         FOR I := 1 TO NROWS DO
            BEGIN
               WRITELN (I:5,STEND[I,1]:10,STEND[I,2]:10)
            END;
      WRITELN;
      WRITELN ('         DROP-IN LOCATIONS');
      WRITELN ('  NO.', '       ROW', '    COLUMN');
        FOR I := 1 TO LDROPS DO
           BEGIN
              WRITELN (I:5,DROPS[I,1]:10,DROPS[I,2]:10)
```

```
        END;
  END;


PROCEDURE FILEPID (VAR F1,F2,F3 : TEXT);
  BEGIN
    REWRITE (F1,NAMEPID);
    WRITELN (F1, ' IDENTIFICATION,1034XD');
    WRITELN (F1, ' DIE', DIESIZE[1],DIESIZE[2]);
    WRITELN (F1,' REFERENCE', REFERENCE[1],REFERENCE[2]);
    NUMBER := LOCATION;
    WRITELN (F1,' NUMBER', NUMBER);
    WRITELN (F1, ' END');
      FOR I := 1 TO NUMBER DO
        BEGIN
          WRITELN (F1,' LOCATION',I,XLOC[I],YLOC[I])
        END;
    WRITELN (F1,' END');
    CLOSE (F1);

    REWRITE (F2,NAMEEXT);
      FOR I := 1 TO NROWS DO
        BEGIN
          WRITELN (F2, STEND[I,1],STEND[I,2])
        END;
      WRITELN (F2,ZERO,ZERO);

      IF (LDROPS > 0) THEN
        BEGIN
          FOR I := 1 TO LDROPS DO
            BEGIN
              WRITELN (F2,DROPS[I,1],DROPS[I,2])
            END;
        END;
      WRITELN (F2,ZERO,ZERO);
      CLOSE (F2);

    REWRITE (F3,NAMEPDF);
    WRITELN (F3, ' 1034XD');
    RDIE[1] := DIESIZE[1];
    RDIE[2] := DIESIZE[2];
    RREF[1] := REFERENCE[1];
    RREF[2] := REFERENCE[2];
    RTEMP := ZERO;
    NTEMP := ZERO;
      FOR I := 1 TO NUMBER DO
        BEGIN
          RXLOC[I] := XLOC[I];
          RYLOC[I] := YLOC[I]
        END;
    WRITELN (F3,ONE:14,NUMBER:5, RDIE[1]:7:0,RDIE[2]:7:0,
                  RREF[1]:7:0,RREF[2]:7:0);
    WRITELN(F3,ZERO:14,RTEMP:13,RTEMP:13,RTEMP:13);
      FOR I := 1 TO NUMBER DO
        BEGIN
          WRITELN (F3,I:14,RXLOC[I]:7:0,RYLOC[I]:7:0)
        END;
    CLOSE (F3);
  END;
```

```
BEGIN

BLANK := ' ';
PIDEXT:='.PID';
SADEXT:='.SAD';
PDFEXT:='.PDF';
DISKNAME:='FW2:';


  WRITELN(' ENTER DIE SIZE, 2 INTEGERS IN MACHINE UNITS');
  WRITELN(' X(=WIDTH),  Y(=HEIGHT)');

  READLN (DIESIZE[1],DIESIZE[2]);

  WRITELN (' ENTER REFERENCE DIE POSITION, (COLUMN,ROW), 2 INTEGERS');
  READLN (REFERENCE[1],REFERENCE[2]);

  WRITELN (' ENTER STEND[I,1], STEND[I,2], 2 INTEGERS PER LINE');
  WRITELN (' TYPE 0 0 TO END INPUT TO STEND ARRAY');

     I :=  1;




     100:
         WRITE (' ROW ', I:2, '  >  ');   READ (STEND[I,1],STEND[I,2]);
         IF (STEND[I,1] < 0) OR (STEND[I,2] < 0)  THEN
                 FLAG := 2
         ELSE
            IF (STEND[I,1] = 0) THEN
                  FLAG := 4
            ELSE
                  IF (STEND[I,1] >= STEND[I,2]) THEN
                       FLAG := 1
                  ELSE
                     IF (STEND[I,2] > LSTEN) THEN
                           FLAG := 3
                     ELSE
                           FLAG := 5;
         IF (FLAG <= 3) THEN

            CASE FLAG OF
               1:
                   BEGIN
                        WRITELN (' *** STEND[I,1] .GE. STEND[I,2]');
                        GOTO 100
                   END;
               2:
                   BEGIN
                        WRITELN (' *** NEGATIVE STEND VALUE');
                        GOTO 100
                   END;
               3:
                   BEGIN
                        WRITELN (' *** STEND[I,2] > 32');
```

```
                GOTO 100
            END;
        END;

    IF (FLAG = 5) THEN
       BEGIN
          I := I+1;
          IF ( I <= LSTEN) THEN
              GOTO 100
          ELSE
             BEGIN
                WRITELN (' *** ROW 32 ENTERED, STEND COMPLETE');
                NROWS :=32;
                GOTO 250
             END;
       END;
    NROWS := I-1;
    IF (NROWS = 0) THEN
       BEGIN
          WRITELN (' *** NO FILES CREATED');
          GOTO 900
       END;




250:
     NCOLS := 0;
     FOR I := 1 TO NROWS DO
        BEGIN
           IF (STEND[I,2] > NCOLS) THEN
              NCOLS := STEND[I,2]
        END;
     WRITELN (' ENTER COORDINATES OF DROP-IN SITES, (COLUMN,ROW), 2 INTEGERS PER LINE');
     WRITELN (' TYPE 0 0 TO END INPUT TO DROPS ARRAY');
     I := 1;




300:
    WRITE (' DROP-IN  ', I:2, '  >  ');   READ (DROPS[I,2],DROPS[I,1]);
    IF (DROPS[I,2] = 0) THEN
                        GOTO 350;
    IA := DROPS[I,1];
    JA := DROPS[I,2];
            IF (IA > NROWS) OR (IA < 0)  THEN
                BEGIN
                    WRITELN (' *** INCORRECT DROP IN ROW VALUE');
                    GOTO 300
                END;
            IF (JA < STEND[IA,1]) OR (JA > STEND[IA,2]) THEN
               BEGIN
                  WRITELN (' *** INCORRECT DROP-IN COLUMN VALUE');
                  GOTO 300
               END;
            IF (I = 1) THEN
               BEGIN
```

```
                I := I+1;
                GOTO 300
             END;
          KL :=I-1;
                FOR K := 1 TO KL DO
                   BEGIN
                      IF ((DROPS[K,1] = DROPS[I,1]) AND
                         (DROPS[K,2] = DROPS[I,2])) THEN
                          BEGIN
                             WRITELN (' *** DUPLICATE DROP-IN COORDINATES');
                             GOTO 300
                          END;
                   END;
          I := I+1;
          IF (I <= NDROPS) THEN
             GOTO 300
          ELSE
             WRITELN (' *** DROP-IN 20 ENTERED');




  350:
      LDROPS := I-1;
      PRINTINFO;

N:=9;

FOR I:=1 TO 20 DO
    BEGIN
       NAMEPID[I]:=BLANK;
       NAMEEXT[I]:=BLANK;
       NAMEPDF[I]:=BLANK
    END;

FOR I:=1 TO N DO
    FILENAME[I]:=BLANK;

WRITELN (' ENTER FILENAME ONLY (9 OR LESS CHARACTERS) FOR');
WRITELN ('       .PID, .PDF, AND .SAD FILES');

READLN (FILENAME);

I:=0;
REPEAT
   I:=I+1;
UNTIL (FILENAME[I]=BLANK) OR (I=N);

IF(I<>N) THEN NCHAR:=I-1
ELSE
   NCHAR:=I;

L:=NCHAR+4;

FOR I:=1 TO 4 DO
    NAMEPID[I]:=DISKNAME[I];
FOR I:=1 TO NCHAR DO
    NAMEPID[I+4]:=FILENAME[I];
FOR I:=1 TO 4 DO
```

```
    NAMEPID[I+L]:=PIDEXT[I];

FOR I:=1 TO 4 DO
    NAMEEXT[I]:=DISKNAME[I];
FOR I:=1 TO NCHAR DO
    NAMEEXT[I+4]:=FILENAME[I];
FOR I:=1 TO 4 DO
    NAMEEXT[I+L]:=SADEXT[I];

FOR I:=1 TO 4 DO
    NAMEPDF[I]:=DISKNAME[I];
FOR I:=1 TO NCHAR DO
    NAMEPDF[I+4]:=FILENAME[I];
FOR I:=1 TO 4 DO
    NAMEPDF[I+L]:=PDFEXT[I];


      TALLY := LDROPS;
      FOR K := 1 TO TALLY DO
         BEGIN
            COMPDROPS[K,1] := DROPS[K,1];
            COMPDROPS[K,2] := DROPS[K,2]
         END;
      LOCATION := 0;
      I := 1;




  400:
      X := STEND[I,1];
      LASTCOL := STEND[I,2];




  500:
      IF (TALLY = 0) THEN
         GOTO 700
      ELSE
         K := 1;




  600:
      DROWS := COMPDROPS[K,1];
      DCOLS := COMPDROPS[K,2];
      IF (I = DROWS) AND (X = DCOLS) THEN
         BEGIN
            TALLY := TALLY-1;
            FOR J := K TO TALLY DO
               BEGIN
                  COMPDROPS[J,1] := COMPDROPS[J+1,1];
                  COMPDROPS[J,2] := COMPDROPS[J+1,2]
               END;
            GOTO 800
```

```
            END
         ELSE
            IF (K < TALLY) THEN
               BEGIN
                  K := K+1;
                  GOTO 600
               END
            ELSE




      700:
         LOCATION := LOCATION+1;
         YLOC[LOCATION] := I;
         XLOC[LOCATION] := X;




      800:
         IF ( X < LASTCOL) THEN
            BEGIN
               X := X+1;
               GOTO 500
            END
         ELSE
            IF ( I< NROWS) THEN
               BEGIN
                  I := I+1;
                  GOTO 400
               END
            ELSE
               FILEPID (PIDFILE,EXTFILE,PDFFILE);
      900:
END.
```

APPENDIX II

Sample Run of PCFG Program

```
>RUN PCFG
 ENTER DIE SIZE, 2 INTEGERS IN MACHINE UNITS
 X(=WIDTH),  Y(=HEIGHT)
5430 5430
 ENTER REFERENCE DIE POSITION, (COLUMN,ROW), 2 INTEGERS
4 1
 ENTER STEND[I,1], STEND[I,2], 2 INTEGERS PER LINE
 TYPE 0 0 TO END INPUT TO STEND ARRAY
 ROW  1  >  4 10
 ROW  2  >  1 12
 ROW  3  >  1 12
 ROW  4  >  4 10
 ROW  5  >  0 0
 ENTER COORDINATES OF DROP-IN SITES, (COLUMN,ROW), 2 INTEGERS PER LINE
 TYPE 0 0 TO END INPUT TO DROPS ARRAY
 DROP-IN   1  >  0 0
    NROWS=  4    NCOLS= 12
DIE SIZE= 5430 5430
REFERENCE=    4    1

    I     STEND[I,1]     STEND[I,2]
    1         4              10
    2         1              12
    3         1              12
    4         4              10

         DROP-IN LOCATIONS
 NO.        ROW    COLUMN
ENTER FILENAME ONLY (9 OR LESS CHARACTERS) FOR
      .PID, .PDF, AND .SAD FILES
SAMPLE
>
```

Appendix III

Stat2 Version 2.00A Commands

Following is an alphabetical list of legal commands for Version 2.00A:

AIP - Alter an individual point
ASG - Assign input data file
AXP - Alter excluded points
DIS - Display distribution
END - Terminate STAT2 execution
ENN - Set N to a specified value
ERM - Error message switch
HIS - Draw a histogram
IIP - Include an individual point
LAP - List all points
LIP - List an individual point
LNS - List points beyond N*SIGMA from mean
LXP - List excluded points
MAC - Execute command macro
MP4 - Circular shaded map
PAU - Pause STAT2 execution
PLT - Draw character display of DATA array
PRS - Print statistics
REA - Read input data file
REM - Set or reset remote mode
RES - Restore all points to included status
XGT - Exclude points greater than a value
XIP - Exclude an individual point
XLT - Exclude points less than a value
XNS - Exclude points beyond N*SIGMA from mean
XOL - Exclude outliers
XPP - Exclude peripheral points
XSS - Exclude small sample

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 85/3292	2. Performing Organ. Report No.	3. Publication Date January 1986

4. TITLE AND SUBTITLE

Release Notes for STAT2 Version 2.00A: An Addendum to NBS Special Publication 400-75

5. AUTHOR(S)

C. H. Ellenwood and R. L. Mattis

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

STAT2 is a FORTRAN program which is used to analyze and display data from microelectronic test structures fabricated on semiconductor wafers. The program reads data as a two-dimensional array, extracts sample statistical values, identifies outliers, calculates replacement values for outliers, and makes histograms and circular gray-tone data maps. Version 2.00A is an adaptation of STAT2 to run under Version 3.2 of the RSX-11M operating system. This operating system is used on the automatic tester which acquires the test structure data. Data can then be taken and analyzed on the same system.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

ATE; computer program; outlier; process validation wafer; statistical analysis; two-dimensional map; wafer map

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

18

15. Price

$9.95

USCOMM-DC 6043-P80

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date: Yr. 85 Mo. 12 Day 11

02. Summary prepared by (Name and Phone): Colleen Ellenwood (301) 921-3801

03. Summary action: New [X] Replacement [ ] Deletion [ ]

04. Software date: Yr. 85 Mo. 12 Day 11

05. Software title: Release Notes for STAT2 Version 2.00A: An Addendum to NBS Special Publication 400-75

Previous Internal Software ID:

06. Short title:

07. Internal Software ID: STAT2

08. Software type:
- [ ] Automated Data System
- [X] Computer Program
- [ ] Subroutine/Module

09. Processing mode:
- [X] Interactive
- [ ] Batch
- [ ] Combination

10. Application area:

General:
- [ ] Computer Systems Support/Utility
- [X] Scientific/Engineering
- [ ] Bibliographic/Textual
- [ ] Management/Business
- [ ] Process Control
- [ ] Other

Specific: Analysis of Integrated Circuit Process Validation Wafer Test Data

11. Submitting organization and address:
Semiconductor Electronics Division
National Bureau of Standards
Gaithersburg, MD 20899

12. Technical contact(s) and phone:
Colleen H. Ellenwood
(301) 921-3801

13. Narrative:
The program reads data as a two-dimensional array, extracts sample statistical values, identifies outliers, calculates replacement values for outliers, and makes histograms and circular gray-tone data maps. Version 2.00A is an adaptation of STAT2 to run under Version 3.2 of the RSX-11M operating system. This operating system is used on the automatic tester which acquires the test structure data. Data can therefore be taken and analyzed on the same system.

14. Keywords:
ATE; computer program; outlier; process validation wafer; statistical analysis; two-dimensional map; wafer map

15. Computer manuf'r and model	16. Computer operating system	17. Programing language(s)	18. Number of source program statements
DEC LSI 11/23	RSX-11M 3.2	FORTRAN	6364

19. Computer memory requirements	20. Tape drives	21. Disk/Drum units	22. Terminals
30kB	None	Disk needed for map scratch file, data input	1 video or hard copy

23. Other operational requirements:

24. Software availability:
- Available [X]
- Limited [ ]
- In-house only [ ]

25. Documentation availability:
- Available [X]
- Inadequate [ ]
- In-house only [ ]

The user's manual is available and consists of NBS Spec. Publ. 400-75, NBSIR 83-2779, and NBSIR 85-3292

26. FOR SUBMITTING ORGANIZATION USE:

Standard Form 185
1974 July
U.S. Dept. of Commerce—NBS
(FIPS. Pub. 30)

INSTRUCTIONS

01. Summary Date. Enter date summary prepared. Use Year, Month, Day format: YYMMDD.

02. Summary Prepared By. Enter name and phone number (including area code) of individual who prepared this summary.

03. Summary Action. Mark the appropriate box for new summary, replacement summary or deletion of summary. If this software summary is a replacement, enter under "Previous Internal Software ID" the internal software identification as reported in item 07 of the original summary, and enter the new internal software identification in item 07 of this form; complete all other items as for a new summary. If a software summary is to be deleted, enter under "Previous Internal Software ID" the internal software identification as reported in item 07 of the original summary; complete only items 01, 02, 03 and 11 on this form.

04. Software Date. Enter date software was completed or last updated. Use Year, Month, Day format: YYMMDD.

05. Software Title. Make title as descriptive as possible.

06. Short Title. (Optional) Enter commonly used abbreviation or acronym which identifies the software.

07. Internal Software ID. Enter a unique identification number or code.

08. Software Type. Mark the appropriate box for an **Automated Data System** (set of computer programs), **Computer Program**, or **Subroutine/Module**, whichever best describes the software.

09. Processing Mode. Mark the appropriate box for an **Interactive**, **Batch**, or **Combination** mode, whichever best describes the software.

10. Application Area.

General: Mark the appropriate box which best describes the general area of application from among:

Computer Systems Support/Utility	**Process Control**
Management/Business	**Bibliographic/Textual**
Scientific/Engineering	**Other**

Specific: Specify the sub-area of application; e.g.: "COBOL optimizer" if the general area is "Computer Systems Support/Utility"; "Payroll" if the general area is "Management/Business"; etc. Elaborate here if the general area is "Other."

11. Submitting Organization and Address. Identify the organization responsible for the software as completely as possible, to the Branch or Division level, but including Agency, Department (Bureau/Administration), Service, Corporation, Commission, or Council. Fill in complete mailing address, including mail code, street address, city, state, and ZIP code.

12. Technical Contact(s) and Phone: Enter person(s) or office(s) to be contacted for technical information on subject matter and/or operational aspects of software. Include telephone area code. Provide organization name and mailing address, if different from that in item 11.

13. Narrative. Describe concisely the problem addressed and methods of solution. Include significant factors such as special operating system modifications, security concerns, relationships to other software, input and output media, virtual memory requirements, and unique hardware features. Cite references, if appropriate.

14. Keywords. List significant words or phrases which reflect the functions, applications and features of the software. Separate entries with semicolons.

15. Computer Manufacturer and Model. Identify mainframe computer(s) on which software is operational.

16. Computer Operating System. Enter name, number, and release under which software is operating. Identify enhancements in the Narrative (item 13).

17. Programing Language(s). Identify the language(s) in which the software is written, including version; e.g., ANSI COBOL, FORTRAN V, SIMSCRIPT II.5, SLEUTH II.

18. Number of Source Program Statements. Include statements in this software, separate macros, called subroutines, etc.

19. Computer Memory Requirements. Enter minimum internal memory necessary to execute software, exclusive of memory required for the operating system. Specify words, bytes, characters, etc., and number of bits per unit. Identify virtual memory requirements in the Narrative (item 13).

20. Tape Drives. Identify number needed to operate software. Specify, if critical, manufacturer, model, tracks, recording density, etc.

21. Disk/Drum Units. Identify number and size (in same units as "Memory"—item 19) needed to operate software. Specify, if critical, manufacturer, model, etc.

22. Terminals. Identify number of terminals required. Specify, if critical, type, speed, character set, screen/line size, etc.

23 Other Operational Requirements. Identify peripheral devices, support software, or related equipment not indicated above, e.g., optical character devices, facsimile, computer-output microfilm, graphic plotters.

24. Software Availability. Mark the appropriate box which best describes the software availability from among: **Available to the Public**, **Limited Availability** (e.g.: for government use only), and **For-In-house Use Only.** If the software is "Available", include a mail or phone contact point, as well as the price and form in which the software is available, if possible.

25. Documentation Availability. Mark the appropriate box which best describes the documentation availability from among: **Available to the Public**, **Inadequate for Distribution**, and **For In-house Use Only.** If documentation is "Available", include a mail or phone contact point, as well as the price and form in which the documentation is available, if possible. If documentation is presently "Inadequate", show the expected availability date.

26. For Submitting Organization Use. This area is provided for the use of the organization submitting this summary. It may contain any information deemed useful for internal operation.

GPO : 1974 O - 544-655